Google Glass au regard de son API : visions & régulations d'une innovation technique

Samuel Goyet

Études Littéraires, Université Laval, Québec ; Sciences de l'Information et de la Communication, GRIPIC, Paris-Sorbonne.
Mail : samuel.goyet@gmail.com

Résumé : Si les *Google Glasses* ne sont pas encore disponibles au grand public, Google en a déjà fourni l'interface de programmation (API) permettant d'écrire des applications informatiques pour ce nouveau dispositif. À travers cette API, quelle vision du service est donnée ? Quelles valeurs y sont affirmées et comment est guidée l'appropriation de cette apparente innovation technologique ? Une analyse sémiotique du site https://developers.google.com/glass/ permettra de mieux comprendre le discours d'escorte de Google Glass.

Mots-clés : Google – Interface – Innovation – discours d'escorte – design – sémiotique – écrits d'écran - API

1. Historique des Google Glass

Le 5 avril 2012, à l'occasion d'un évènement de charité, Sergueï Brin, le co-fondateur de Google et directeur du Google X-Labs, faisait la première démonstration publique des Google Glass. Ce qui n'était à l'époque qu'un prototype alimenta rapidement les spéculations quant à ce qu'allait être le futur des technologies numériques, entre réalité augmentée et *wearable technologies*. Deux mois plus tard, le 27 juin 2012, lors de la conférence Google I/O, le « Projet Explorateur » est lancé, permettant à certains développeurs triés sur le volet d'essayer les premiers prototypes pour 1500$. Le 15 avril 2013, ce programme est élargi et une API (la *Mirror API*) est ouverte aux sélectionnés, puis au grand public le 25 novembre. En février 2014, devant de nombreux problèmes suite à l'utilisation des Google Glass dans l'espace public, Google publie

adfa, p. 1, 2011.
© Springer-Verlag Berlin Heidelberg 2011

une charte d'utilisation de son produit[1]. Enfin, le programme Explorateur est ouvert à tous lors d'une « vente flash » des Google Glass le 15 avril 2014.

2. Objet technique et discours d'accompagnement [1]

2.1. Accompagner la nouveauté

Au-delà du lancement marketing de ce produit, force est de constater que les Google Glass concentrent aujourd'hui un ensemble d'imaginaires à la fois technophiles[2] et technophobes[3], comme d'autres chercheurs ont pu le constater par ailleurs sur d'autres technologies de l'information et de la communication [2]. Comme souvent dans le cas de l'innovation technique, une rhétorique de la nouveauté est convoquée par ses concepteurs, pour mieux souligner la « rupture radicale » que constituerait le produit [3]. Si les Google Glass posent effectivement des questions essentielles sur l'exploitation économique des *wearable technologies*, des types de données ainsi générées et sur leur application ou bien encore sur la recomposition de la notion de vie privée [4], nous aimerions nous focaliser sur la manière dont Google, à travers l'API *Mirror* et d'autres textes normatifs, contrôle la diffusion de cette technologie dans la société, en élaborant sur son produit un discours axiologique à destination des utilisateurs et des développeurs.

2.2. L'API comme composite [5 ; 6] socio-technique

En son sens strictement technique, une API, ou interface de programmation, peut être définie comme la formalisation, en termes informatiques, de l'accès aux données ou ressources détenues par une entreprise ou une institution, permettant l'utilisation de ces données par des développeurs et/ou un logiciel tiers. L'API *Mirror* est une API web publique, c'est-à-dire reposant sur le protocole HTTP pour structurer les demandes de données, ou "requêtes", et étant publiée sur un site web. Sous réserve d'authentification auprès du détenteur des données, un développeur extérieur peut donc utiliser ces données pour créer une application inédite, en tout cas non prévue par l'entreprise publiant l'accès à ses données, ici Google. En ce sens, une API est non seulement un élément essentiel guidant le développement d'applications comme « mashups » [7] mais aussi un ressort économique qui promeut une « culture collaborative » du développement logiciel [8]. Plus que cela encore : en ceci qu'elle est pour un développeur la seule modalité d'accès à des données, une API est le lieu d'une régulation basée sur l'écriture informatique [9] : Google y édicte des conditions

[1] https://sites.google.com/site/glasscomms/glass-explorers
[2] Voir par exemple la communauté http://www.goglasses.fr
[3] http://www.washingtonpost.com/business/technology/anti-glass-attack-in-san-francisco-highlights-tension-over-wearables/2014/02/26/b3f21e44-9eeb-11e3-9ba6-800d1192d08b_story.html

d'utilisation, des recommandations de design, un discours sur son produit, sa visée… [10]. Ce discours est lisible à travers la documentation de l'API, qui est à la fois une véritable grammatisation [11] des différentes composantes informatiques du produit et une pratique lettrée, au sens où cette documentation exerce un « double contrôle » [12] sur l'écriture de l'application à venir : contrôle sur le contenu (formulation correcte des requêtes, authentification en bonne et due forme…) et sur la forme de l'application (recommandation de design d'interface). La Mirror API des Google Glass sera donc analysée comme composite [5,6], dessinant un ensemble d'injonctions et/ou de prescriptions sur ce produit. Autrement dit, comment à travers son API dédiée, Google prépare-t-il l'adoption de son produit ? Qu'est-ce que l'API des Google Glass dit de et sur Google Glass en tant qu'objet socio-technique encore largement fantasmé, puisque quasiment pas disponible pour le grand public ?

3. Méthodologie

Nous ferons une analyse sémiotique du site https://developers.google.com/glass/, en suivant le cadre conceptuel et méthodologique de la sémiotique des écrits d'écran [13], permettant de décrire et d'analyser précisément comment le site hébergeant l'API *Mirror* construit un discours autour des Google Glass. La théorie des écrits d'écran permet en effet de voir dans la construction d'un discours d'escorte le rôle structurant des formes et pratiques d'écritures liées à la culture des développeurs (documentation, manuels, consoles test…) mais aussi de prendre en compte les spécificités du fonctionnement technique des médias informatisés, reposant sur une « double écriture » [14, 15] : les procédures algorithmiques définies par le code d'un côté ; la conception de l'interface graphique de l'autre. Au sein de ces couches d'écriture, nous proposons de considérer qu'un API fonctionne à trois niveaux : le niveau paramétrique, qui définit la grammaire des requêtes possibles ; le niveau pragmatique, soit la manière dont un développeur va selon son projet d'application utiliser cette grammaire ; le niveau (provisoirement appelé) sémiotique, soit celui de l'interface grand public du site ou de l'application qui utilise l'API. Ces trois niveaux ne sont pas chronologiquement ordonnées, mais en constante interaction : le paramétrique définit les possibilités sémiotiques, quand le développeur au niveau pragmatique opère des choix et des ajustements en fonction des deux autres niveaux. En raison de ce cadre d'analyse, nous nous concentrerons sur la documentation à proprement parler de l'API *Mirror*, et non pas sur le Kit de Développement Logiciel (SDK) qui ne repose pas exactement sur les mêmes principes techniques et ne met pas en jeu les mêmes enjeux de régulation.

4. Principaux résultats

Un premier point tient au paradoxe entre un discours de la « révolution » et les conditions effectives de son appropriation par les développeurs et les utilisateurs. En dépit du discours promotionnel tenu par Sergueï Brin vantant la nouveauté radicale des Google Glass, l'API de ces dernières minimise cette nouveauté : les manières

de programmer seront similaires aux autres produits Google, les langages également… C'est donc un point de vue techno-économique qui est adopté ici, qui tient au processus de fabrication des logiciels et à leur commercialisation. L'enjeu est au moins triple : rassurer les développeurs externes amenés à utiliser l'API ; afin de faciliter le développement de logiciels, source de revenus pour Google ; tout en réaffirmant la filiation des Google Glass à la firme Google, tant du point de vue technique que symbolique.

La distinction entre *Design*, *Develop* & *Distribute* structure entièrement le site et la documentation de l'API. Elle n'est pas neutre, puisqu'elle est une modélisation du processus de développement logiciel : d'abord concevoir l'interface utilisateur de l'application, ensuite en écrire le code correspondant pour enfin rendre disponible cette application *via* le magasin en ligne de Google. En apparence donc, Google à travers l'API *Mirror* reprend un modèle classique de développement logiciel en y ajoutant une finalité marchande qui le sert directement. Une analyse plus précise tempère ce constat initial, tant la notion d'"expérience utilisateur" est constamment mentionnée comme mètre étalon du design et du développement des applications pour Google Glass. Tant et si bien que l'"expérience utilisateur" remplit une fonction téléologique : elle doit être la finalité ultime de chacune des étapes de fabrication du logiciel. Nous arrivons donc à la co-présence de deux modes de finalité : l'une chronologique (la distribution marchande), l'autre qui tient quasiment d'un impératif esthétique et éthique (privilégier l'"expérience utilisateur"). Notre hypothèse est qu'en convoquant cette figure d'un utilisateur abstrait, voire fantasmé, Google réaffirme encore plus son contrôle sur la chaîne de fabrication des logiciels pour son nouveau produit. Contrôlant de fait la distribution, le problème réside principalement dans la maîtrise de la façon dont les programmes sont écrits et conçus. D'où l'anticipation, au coeur du niveau paramétrique de l'API, de son niveau pragmatique,à travers la figure de l'utilisateur. Cette dernière est en quelque sorte une charnière, un mécanisme de régulation entre la syntaxe de ce qui peut être programmé grâce à l'API et les programmes qui seront effectivement écrits.

Nous finirons par analyser encore plus en détail cette dyade *Design* / *Develop*. Car ce qu'elle traduit, c'est la dynamique fondamentale des médias informatisés en tant qu'ils reposent sur une double écriture qui correspond au niveau paramétrique et sémiotique de l'API, En choisissant ce chapitrage, Google textualise donc à la fois les procédures informatiques et couches architextuelles en action (par la documentation à proprement parler, l'explicitation de la syntaxe des requêtes…), mais aussi les contextes d'utilisation des Google Glass tels qu'ils sont constructibles à travers des interfaces graphiques. La mise à jour de cette double textualisation permet *in fine* de comprendre comment Google propose dans l'API *Mirror* des "scénarios" [5] où s'entremêlent des considérations techniques, esthétiques et éthiques.

Références bibliographiques

1. Jeanneret, Y.: Autre chose qu'un discours, davantage qu'un accompagnement, mieux qu'une résistance. Terminal 85, 107-117 (2011)
2. Mallein, Ph., Toussaint, Y. : L'intégration sociale des technologies d'information et de communication : une sociologie des usages. Technologies de l'information et Société, vol. 6 n°4, 315-334 (1994)
3. Jeanneret, Y. : Y a-t-il (vraiment) des technologies de l'information ? Presses Universitaires du Septentrion, Villeneuve d'Ascq (2007)
4. Maurel, L. : Les Google Glass peuvent-elles changer le statut juridique de la vision ? http://scinfolex.com/2013/04/12/les-google-glasses-peuvent-elles-changer-le-statut-jurudique-de-la-vision/
5. Akrich, M.: Comment décrire les objets techniques ? Technique et Culture 9, 49-64 (1987)
6. Le Marec, J. : Ce que le « terrain » fait aux concepts : vers une théorie des composites. HDR soutenue sous la direction de Baudouin Jurdant, Université Paris 7 (2002)
7. Benslimane, D., Dustdar, S. & Sheth, A. : Services Mashups. The New Generation of Web Applications. Internet Computing, vol. 12 n°5, 13-15 (2008)
8. Rieder, B. : Entre marché et communauté : une discussion de la culture participative à l'exemple de Google Maps. In : Ludovia 2008 : Do it yourself 2.0., pp. 282-292. Ludovia, Aix-les-Thermes, (2008)
9. Lessig, L. : Code 2.0. Basic Books, New York (2006)
10. Bucher, T. : Objects of intense feeling : the case of the twitter API. Computationnal Culture 3, http://computationalculture.net/article/objects-of-intense-feeling-the-case-of-the-twitter-api (2013)
11. Auroux, S. : La Révolution technologique de la grammatisation. Mardaga, Liège (1994)
12. Jacob, Ch. : La carte des mondes lettrées. In : Giard L. & Jacob Ch. (Dir.), Des Alexandries I. Du livre au texte. pp.11-40. Bibliothèque Nationale de France, Paris (2001)
13. Souchier, E. : L'écrit d'écran, pratiques d'écriture et informatique. Communication & Langages 107, 105-119 (1996)
14. Knuth, D. : Litterate programming. Stanford University, California (1992)
15. Souchier, E. : De la *lettrure* à l'écran. Vers une lecture sans mémoire ? In : Schuerewegen, F (Dir.). Texte, 25-26, pp. 47-68. Université de Toronto, Canada (2000)